Towards Personalized Image Retrieval

Stéphane Bissol, Philippe Mulhem, Yves Chiaramella

Towards Personalized Image Retrieval

Stéphane Bissol[1], Philippe Mulhem[1], Yves Chiaramella[1]

[1] CLIPS-IMAG Laboratory, 385 rue de la Bibliothèque BP53,
38041 Grenoble cedex, France
{Bissol, Mulhem, Chiara}@imag.fr

Abstract. This paper describes an approach to personalized image indexing and retrieval. To tackle the issue of subjectivity in Content-Based Image Retrieval (CBIR), users can define their own indexing vocabulary and make the system learn it. These indexing concepts may be both local (objects) and global (image categories). The system guides the user in the selection of relevant training examples. Concept learning in the system is incremental and hierarchical: global concepts are built upon local concepts as well as low-level features. Similarity measures tuning is used to emphasize relevant features for a given concept. To illustrate the potential of this approach, an implementation of this model has been developed; preliminary results are given in this paper.

1 Introduction

Unorganized collections of documents often lead to time consuming or deficient retrieval. Owning thousands of documents looses its purpose if those cannot be retrieved properly. This is particularly justified regarding the digital images field. Thanks to technological breakthroughs in the digitization, storage and accessibility domains, users may personally possess several thousands of digital images [1]. Among them, rare are those who actually organize their collection into a set of easily retrievable photographs. Each image becomes thus less reachable as the collection grows. In order to make each image retrievable, a set of images must be in some way indexed. Manual indexing is generally inappropriate, requiring too much time and effort. On the other hand (and unlike textual documents), an image semantic content cannot be accessed straightforwardly. Therefore automatic indexing implies, somehow, the decryption of the information hidden in the image pixels. Automatic indexing systems already exist that are able to associate symbolic information to images (like the objects they contain or the category they belong to). However, in those systems, symbolic terms used for indexing are arbitrarily chosen by the designers and may no suit the way users search for their images. The reason is that describing an image is known to be a highly subjective task [2].

This paper describes an automatic indexing system for any type of consistent image database. It allows users to define their own indexing terms progressively and in accordance to their specific needs, be they global (image categories) or local (objects present in the image). Low-level features (as colors and textures) underlying the labeling are hidden from the user when building mid-level features (objects contained

in the image such as “Sky”, “person” or “Foliage”). In the same fashion, low and mid-level features are implicitly manipulated when learning image categories. The system uses instance-based learning and similarity measure fitting to learn from user interactions. The advantages of the presented approach are that only a few user interactions are required to start up the system, the indexing precision increases fast with the size of the training set and that the indexing vocabulary is inherently unlimited.

The following sub-section reviews the current research into Content Based Image Retrieval (CBIR). Section 2 then introduces the general ideas, as well as the system framework. Low-level features extraction is discussed in section 3. The fourth section explains the way training sets are built. In section 5 we show how to optimize similarity measures used to compare images and regions throughout the system. The construction of classifiers is described in section 6, followed by the categories indexing scheme (section 7). Finally, experiments and results are discussed in section 8, before concluding and sketching future work in section 9.

1.1 Content-Based Image Retrieval

Content-based image retrieval (CBIR) systems [3] intend to provide solutions for browsing and retrieval of visual data. Such systems are dedicated to automatically (or semi-automatically) index images and make possible their retrieval via a search engine. Most CBIR systems are based upon the same framework [4]: They usually involve a features extraction module (colors, textures, sometimes shapes), an indexing module (occasionally using machine learning techniques) and a retrieval module. However, they differ in the abstraction level of their image description and their respective query systems (which are actually limited by the abstraction level) and are not equal in terms of user’s cognitive effort requirements. As the level of abstraction increases, formulating a query becomes easier.

In the earlier times, CBIR systems would involve only low-level indexes such as color and texture descriptions. In such a system, queries are intrinsically restricted to low-level queries such as sketch drawing, query by example or explicit manipulation of color, texture or shape features. These approaches are suitable in some particular cases (“I want an image which looks like this one”) but are limited and should rather be available as an alternative query tool. Sketch drawing requires a lot of efforts from the user. Query by example is inherently ambiguous: an image may have been chosen as an example for multiple reasons (an object in the image, all the objects, the overall “look”, the atmosphere, the location etc…). Besides, examples might not be available. QBIC [5], the pioneer CBIR system, uses color, texture and shape features to describe images content. As for the queries, the user is given a choice between drawing a sketch and searching directly in terms of colors, shapes and textures. In such a system the abstraction level is low, which delegates a large part of the work to the user. This is also the case for VisualSEEk [6] or Netra [7].

A higher level of abstraction is reached when associating the whole image with a type or a category. A higher level of indexing naturally allows a higher level query formulation. For instance in Simplicity from Stanford University [8], images are automatically classified into categories such as photography, drawing etc… These categories have been learned previously from training sets and are therefore preset. In [9] vacation images are first classified into indoor/outdoor classes. Outdoor images are then further classified into city/landscape classes and landscape photographs are finally sorted into subsets such as forest, mountain or sunsets. The vacation pictures database enhanced by this system becomes much more organized and searchable. Here again, these categories are arbitrarily fixed and might not cover all of the user needs. Furthermore, only the whole image is taken into consideration, confining the queries to these predefined image categories.

Higher in the abstraction scale are the systems which attempt to describe images in terms of the entities composing them (as well as global descriptors). Not only can these systems associate signal information with symbols, they are also able of segmentation. This enables more precise and intuitive queries involving objects, their respective sizes and their spatial relationships. This level of description is, to date, the highest feasible degree of abstraction since, beyond, systems would have to deal with human emotions (atmosphere of a photograph for instance) or a deep knowledge of the world which generated the images (location, what are the people on the picture doing, etc...).

In [10], the authors describe an indexing/retrieval scheme based on high-level visual categories such as grass, sky or wood. These categories are learnt through a neural network classifier using the back-propagation algorithm. Good results are obtained with classification accuracies ranging from 86% to 98% depending on the visual category. However, the indexing terms are pre-defined and the use of neural networks makes incremental learning troublesome. An interesting fact pointed out in [10] is that results are worst when using amateur home pictures instead of a professional photograph collection such as Corel Photo Library. This should be kept in mind when confronting CBIR systems.

2 System Overview

In the remaining of the paper, the system will be called “Citra”, the name given to its implementation. Citra is mainly an image indexing system but also includes a query and retrieval scheme. It allows a user to incrementally define his own indexing concepts, be they intra-image (“Palm tree”, “Sky”, “Person”) or image categories (“Mountains”, “Indoor”, “Underwater”). After a new concept has been defined, all images in the database are updated. Citra does not need to re-compute all the indexes since each label is managed by a specialized classifier. Each label can therefore be updated separately. Figure 1 shows the principal components of the systems.

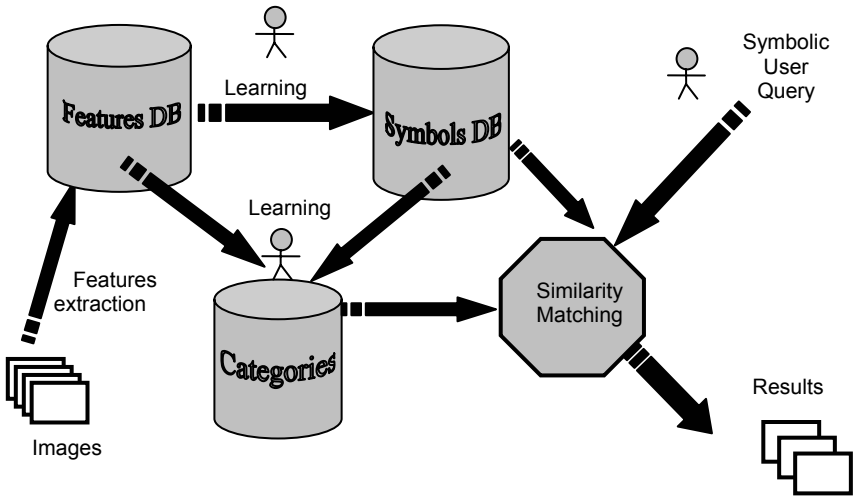

Fig. 1. System overview. Steps requiring user intervention are marked with a symbol.

Two different types of indexing are proposed. The first one is object based and requires the user to select positive and negative examples of the object he/she wishes to define through a semi-automatic segmentation tool (described in section 3). Every example is represented by a set of low-level features extracted from a rectangular, size-standardized region. Prior to building the classifier (section 5), Citra tunes the similarity measure used for each concept by giving weights to features (section 4). The system considers as good a similarity measure able to distinguish positive examples from negative ones. The classifier, a simple KNN classifier uses this similarity measure as well as the positive and negative examples to label unknown regions.

The second indexing concerns image categories (section 6). In the same fashion the user selects a few images corresponding either to positive or negative examples of an image category. A similar process then computes a suitable similarity measure (between images) and builds a classifier. This time however, the features are not only low-level (image color histogram, textures…) but also high-level features previously defined (objects or image categories). For example, when defining the global indexing concept "Beach Picture" the similarity measure tuning may set the features weights in such a way that the global colors, the presence of sky, sea and sand labels are emphasized.

Note that the underlying mechanism for global and local indexing is the same in both cases. The difference lies in the fact that category indexing uses information about objects composing images in addition to low-level features. We call that hierarchical indexing and use it to achieve higher indexing and retrieval abstraction.

3 Extracting Low-Level Features

There are plenty of models to choose from when it comes to extracting image features. The task is to select those which fit relevant criterions.

Extracting features usually yields several values that may or may not be correlated from each other. This characteristic is relevant in our system because (as we describe in section 5) the algorithm which associates relative weights to features assumes that the latter are not correlated.

Invariance to different geometric and lighting transformations is a property we are looking forward to, since pictures having the same content, modulo these transformations, should be considered as similar.

Driven by these considerations, we have chosen the HSV color space and the histogram representation for the characterization of image colors. The HSV color model fulfills the no-correlation property [11] as well as (to a certain extent) the invariance property.

As for texture, we have implemented a feature extractor (that we will not describe in details here) which focuses mainly on roughness and directionality. Roughness is measured under different scales and directions but expressed with disregard to them to ensure invariance. In the same way, directionality is only characterized by a quantity, intentionally omitting the orientation.

4 Building Training Sets

The composition of a training set is a factor influencing the learning based on this set. However in Citra, the user must constitute himself the training sets and no assumption is made concerning his ability to select the "right elements" with respect to the underlying learning mechanism. Therefore, the user has to be guided by the system.

In this section we describe how we build the training sets used to learn intra-image objects in Citra. Unlike most systems like [9, 10], learning new image categories or objects may be done dynamically by the user. Choosing regions or images to build the training sets is usually performed offline by experts who, besides, know the inner working of the system. Picking up the right elements to constitute the training set is essential, especially when the latter is modest in size [12]. Besides, positive examples should be heterogeneous enough to cover all the possible appearances of an entity. As for negative examples, they ought to be close enough to the positive examples to avoid over-generalization and, at the same time, not too surrounding to prevent from over-fitting [12].

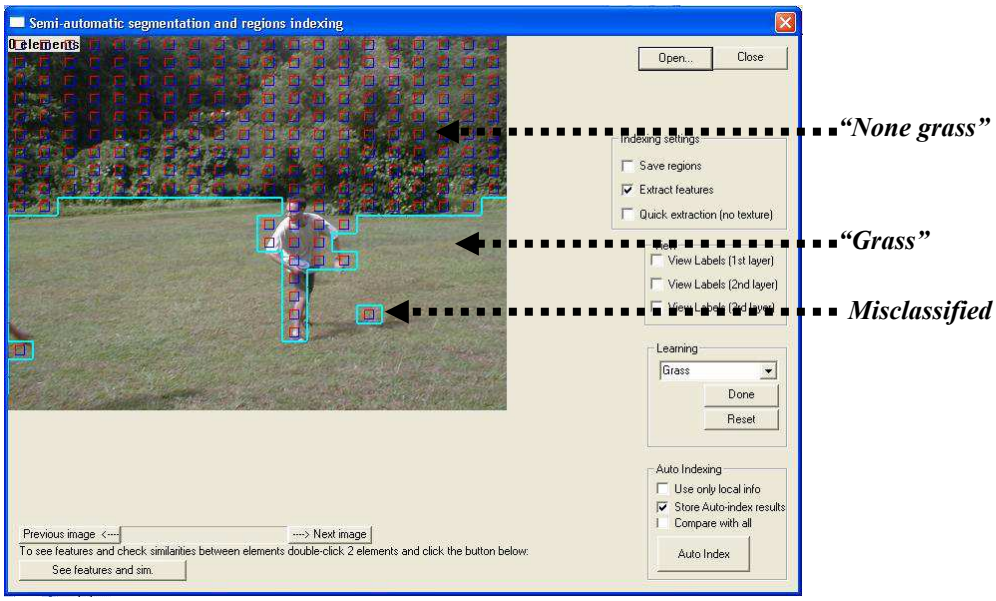

Fig. 2. Selecting relevant positive and negative examples through the semi-automatic segmentation tool.

However in Citra, examples constituting the training set are not selected by an expert but by the user who picks them among a few images. Figure 2 shows the tool proposed to select positive and negative examples. This semi-automatic segmentation tool partition the image into positive and negatives regions after each user click. For instance, when defining "grass" like in figure 2, the user left-clicks in the grass area (defining an example) then in a "none grass" area (defining a negative example). The system segments the image into "grass" and "none grass" areas. In case some areas are misclassified, the user corrects the segmentation by left or right-clicking in those areas, until are no errors are left. That way, the user is imperceptibly forced to select relevant and discriminant examples. This is essential since the training set, manually constituted by the user, may be very small. In order to ensure consistency, the semi-automatic segmentation uses the same low-level features and the same classifiers as the indexing process.

5 Fine-Tuning Similarity Measures

In machine learning, and here more specifically, in empirical learning, one may either attempt to reach a fine level of representation and modeling of the data or concentrate on the problem of how to compare entities. The former approach fits, to a greater degree, problems in which the concepts to be learnt are well defined and formatted. Training examples and concepts chosen by an expert are more likely to be manageable by a machine learning algorithm. In Citra, those are defined by the user

for reasons that might not be obvious. Therefore, fine-tuning similarity measures might be more efficient than attempting to infer fine representations from the data.

Prior to building classifiers, there is a need to understand on which grounds the user based his selection of positive and negative examples. This is equivalent to finding a similarity measure between examples which emphasizes the right features. For instance, when learning the concept "Sky", features describing colors (or even blue colors) might be interesting features to focus on. While if "Car" is the target concept, a similarity measure giving more weight to textures or shapes would probably achieve better results than colors. These weights have to be discovered automatically. More precisely, the problem is to optimize the similarity measure S used to compare feature vectors. S has the following form:

$$S(v1, v2) = \sum_{i=1}^{N} \alpha_i . \mathrm{dist}(v1_i, v2_i) \quad (1)$$

Where N is the number of features, v1 and v2 are the 2 features vectors corresponding to the description of two images, α_i are the weights assigned to the features i. Let k be the number of values than each α_i can take. The number of different weight vectors is k^N, therefore optimizing S is, in the general case, an NP-hard optimization problem. Optimizing S by searching comprehensively the solution space is hence unfeasible.

This is the reason why we make the assumption that features are not correlated with each other. That way, it is possible to assess the *'utility'* of a given feature (for a given concept) independently, which is feasible.

Let us first define what we call utility: Utility is the ability of a given feature to discriminate positive from negative examples of a concept, that is, seeing positive examples as similar and negative examples as dissimilar from positive ones.

For this, we compute the "intra-class" similarity and "inter-class" similarity using a similarity measure which only takes into account the feature under examination. Let E_+ be any positive example and E_- any negative example and $S(E_1, E_2)$ the similarity between two examples. Intra and inter-class similarities are defined as:

$$S_{intra} = \frac{\sum_{i=1}^{P_+ - 1} \sum_{j=1+1}^{P_+} S(E_+^i, E_+^j)}{\frac{P_+^{\,2}}{2} - P_+} \quad (2)$$

$$S_{inter} = \frac{\sum_{i=1}^{P_+} \sum_{j=1}^{P_-} S(E_+^i, E_-^j)}{P_+ P_-} \quad (3)$$

Where P_+ and P_- are respectively the cardinal of the positive examples set and negative examples set.

Then, we combine these two values to get the utility:

$$U = \frac{\lambda_1 D\text{intra}}{\lambda_2 D\,\text{inter}} \quad (4)$$

Utility is computed for each feature. All these values are then normalized and constitute the weights vector for a particular concept. Note that, just like the tf/idf information retrieval classic weighting scheme, features having similar values in all the examples will not be emphasized whereas features having different values in positive and negative examples will be given more weight.

6 Classifiers & Indexing

Once a new concept has been defined by the user (by selecting positive and negative examples of it) and its associated similarity measure has been computed, the system classifies all the regions of all the images according to this new concept. In Citra, membership to a given class is assessed by a specific classifier. Among the multiple approaches, we have chosen to use an instance-based method: the k-Nearest Neighbor (Knn) classifier [12].

Function-fitting methods construct an explicit description of a target function from learning examples, instance-based methods store the training examples. Knn classifiers do not have a theoretical limit to the target function's complexity they learn [12]. This is required in Citra since the target function is determined by the user. Whatever the function-fitting algorithm may be, a set of training example selected by the user exists that will bewilder the classifier. Knn tackles this problem by *not* fitting a function at all. This makes the system flexible and adaptive to the user needs and to the images content.

One of the main issues when dealing with Knn classifiers is to cope with irrelevant attributes, which dilute relevant attributes influence. The similarity measure adapting scheme described in section 5 undertakes this difficulty by reducing the irrelevant attributes relative weights. How to select meaningful training examples is another issue in Knn learning [12], our selection scheme which handles this problem has been discussed in section 4.

7 Indexing Image Categories

In Citra the user may define image categories such as “Under-water”, “City-scape”, or “Indoor”. In addition to our previous works [13], an image is indexed by low-level features as well as high-level features (see figure 3).

Feature	Hue Histogram	Saturation Histogram	Orientation Histogram	…	Trees %	Sand %	Building%	…	Indoor Picture	Beach Picture
Type	Histogram	Histogram	Histogram	…	[0,1]	[0,1]	[0,1]	…	[0,1]	[0,1]
Value	H1	H2	H3	…	0.73	0.12	0.0	…	0.05	0.96

Fig. 3. Feature vector for an image, containing both signal and semantic aspects.

In the feature vector, low-level features (principally histograms) are global to the image and concern color and texture. High-level features are either the *area covered* by a given learnt object (“Sky”, “Trees”) or the *membership probability* to an image category (“Indoor”, “Portrait”).

As discussed in [13], global descriptors are useful but not sufficient to learn an image category. Introducing high-level descriptors greatly improves learning since an image category is strongly linked to the objects contained in the image as well as other image categories. For instance, the category “Mountains” would be connected with “Indoor” (there are no indoor mountains) or “Rocks” (Rocks often occur on a mountain image).

To define a new image category, the user selects a few positive and negative image examples. Similarly to intra-image object definition, the system optimizes a similarity

measure used within a classifier to assign a category membership value to every image in the database.

8. Results

A complete implementation of our model has been realized to conduct experimentations. Images used in the following experiments are personal photographs taken by the authors. These images present a great variability in terms of quality, illumination, locations, type of scene.

Note that the following experimentations are preliminary since we are still in the process of building a large database of manually indexed images for experimentations.

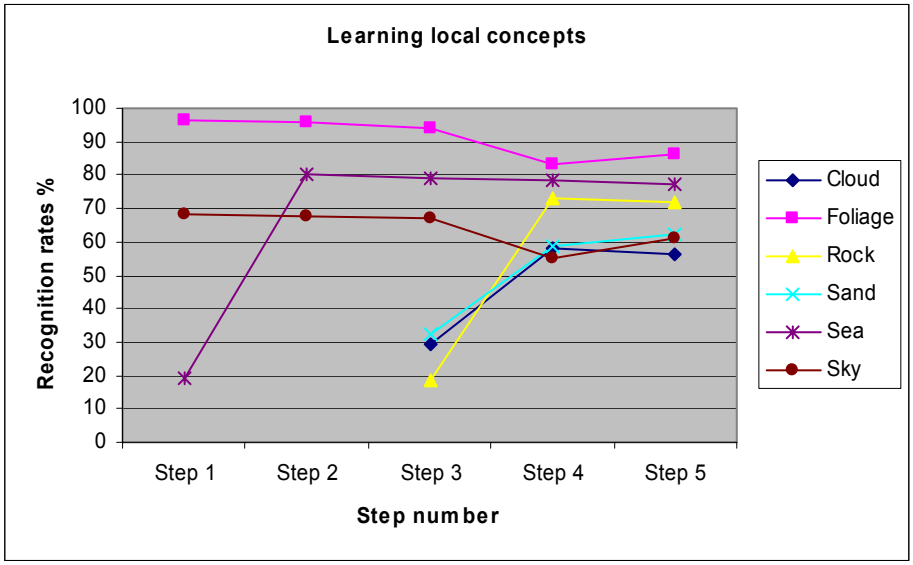

Fig. 4. Recognition results for local concepts over a learning session

8.1 Learning Local Concepts

Figure 4 shows recognition rates for local concepts. The recognition rates are defined here as the *precision* value computed in the field of information retrieval, corresponding to the ratio of the number of correctly classified regions divided by the number of regions classified in that class by the system. This experimentation has been run on

40000 regions (100 images * 400 regions per image). The user starts a session with the system (from scratch) and first defines 3 concepts (step 1) using the semi-automatic segmentation tool on 1 image. The number of positive and negative examples required to segment the image varies depending on the 'complexity' of the concept. In the first step, 3, 3 and 7 examples were required for (respectively) 'Sea', 'Sky' and 'Foliage'.

Noticing many mistakes in the indexing regarding the label 'Sea' the user increases the number of examples in step 2 from 3 to 12, therefore ameliorating the precision.

Then in the next steps, the user introduces new indexing terms (step 3: adding new concepts without modifying the already learnt concepts) and increases the number of examples for concepts whose precision is too low.

There are a few remarks we can make regarding this experiment. Firstly, the introduction of new concepts affects only slightly the precision of other concepts. However, the question remains opened for large number of concepts. Secondly, the accuracy of the indexing increases fast with the training set size, which is a property we were looking forward to. Indeed, the average number of examples per label is 4.3 initially and reaches 14.2 at the last step; meanwhile, the average recognition rate goes from 61% to 70%. Finally, the average precision reached by the system is satisfying, given the small size of the training samples, the large variability of the concepts appearance through the image set and the modest number of low-level features currently extracted.

8.2 Learning Image Categories

Figure 5 shows the result of an experiment in which a user wishes to define a new image category in the system (namely, "Beach pictures"). The lower precision corresponds to the case where only low-level features are used. The addition of symbolic information in the image feature vector (occupation of the image by each local concept, in percents) increases dramatically the precision. This means that global low-level features are not discriminant enough for this category of images, whereas symbolic information is (i.e. 60% 'Sand', 20% 'Sea'). Furthermore we can see that using a weighting scheme for the features is useful as it enhances precision. Note that, for a very small training set, weighting or not the features doesn't affect the precision. This is because finding out the '*utility'* of each feature does not make sense on very small number of examples. However, as the training set size increases the weighting scheme shows its utility by emphasizing the relevant features (i.e. in this case : 'Sand', 'Sea').

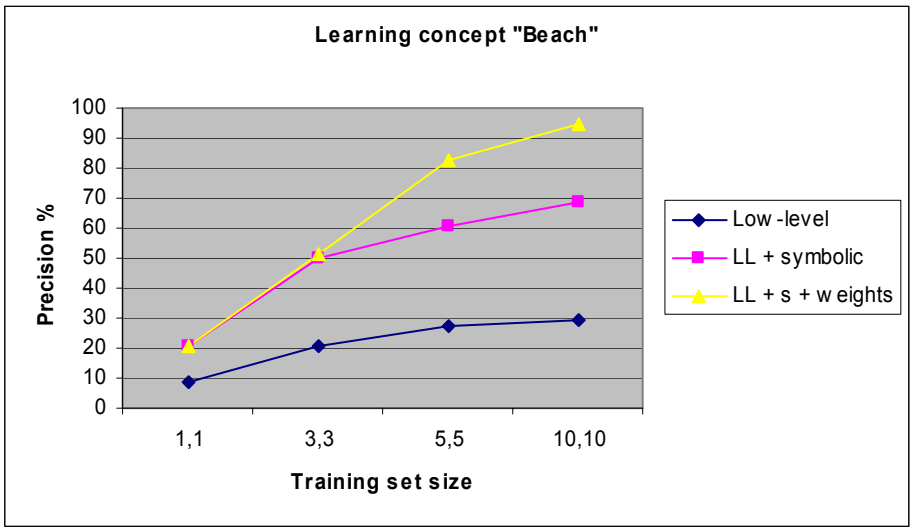

Fig. 5. Precision of the indexing according to the number of examples selected by the user (number of positive examples, number of negative examples).

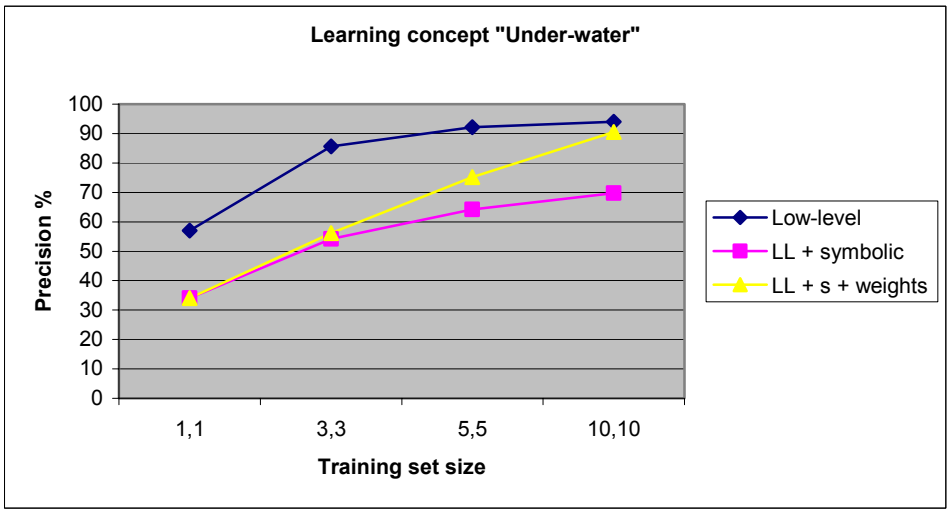

Fig. 6. Precision of the indexing according to the number of examples selected by the user (number of positive examples, number of negative examples).

Figure 6 shows a similar experiment in which the target category is “under-water”. Surprisingly, the introduction here of symbolic information seems to make the preci-

sion drop. The reason is that local concepts that can be found under-water (fishes, shells, etc.) are not known by the system. Therefore, these images are either indexed mostly by the term 'unknown' or by other terms which are mistakes. As a result, the relevant attributes in the feature vectors are diluted, which makes the precision fall.

However, when using the weighting scheme, the precision eventually catches up with the low-level-only precision. This is because irrelevant attributes influence is lessened by their utility value.

9. Conclusion

In this paper we have presented a new user-centered CBIR system. It allows users to define their own indexing terms, be they objects or image categories. Since building a training set is a sensitive task and users of this system are not expected to know how to do it, the system guides them in their selection of examples through a semi-automatic segmentation tool. Additionally, in order to understand on which grounds the user defines concepts, the system is able to characterize the 'utility' of each feature for a given concept (global or local). Users can make the system learn image categories. In this case, the system uses not only signal information but also symbolic information previously learnt: learning is hierarchical.

Our approach gives very promising results. We are currently building a large database of manually indexed images which will allow us to conduct exhaustive experiments.

Since our system performs an efficient feature selection, we plan to add more "perception" to the system via new low-level features, without fearing the dilution of relevant features influence.

Finally, we are currently trying to refine the indexing using statistical information such as location in space of local concepts and co occurrences betweens concepts.

References

1. K. Rodden and K. Wood, How do People Manage Their Digital Photographs?, ACM Conference on Human Factors in Computing Systems Fort Lauderdale, April 2003.
2. *John P Eakins and Margaret E Graham* Content-based Image Retrieval. A report to the JISC Technology Applications Programme. Institute for Image Data Research, University of Northumbria at Newcastle. January 1999
3. A Review of Content-Based Image Retrieval Systems. Colin C. Venters and Dr. Matthew Cooper University of Manchester. www.jtap.ac.uk/reports/htm/jtap-054.html
4. A.W.M. Smeulders, M. Worring, S. Santini, A. Gupta, and R,. Jain, "Content-Based Image Retrieval at the End of Early Years," IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI), vol. 22, no. 12, pp. 1349-1380, Dec. 2000.
5. M. Flickher, H. Sawhney, W. Niblack, J. Ashley, Q. Huang, B. Dom, M. Gorkani, J. Hafner, D. Lee, D. Petkovic, D.Steele, and P. Yanker, "Query by Image and Video Content: The QBIC System," IEEE Computer, vol. 28, no. 9, pp. 23-32, Sept. 1995.

6. J. R. Smith and S.-F. Chang, Querying by color regions using the VisualSEEk content-based visual query system, In Intelligent Multimedia InformationRetrieval. IJCAI, 1996.
7. W.Y.Ma. *NETRA: A Toolbox for Navigating Large Image Databases*. PhD thesis, Dept. of Electrical and Computer Engineering, University of California at Santa Barbara, June 1997.
8. J. Z. Wang, G. Li, and G. Wiederhold. *SIMPLIcity: Semantics-sensitive Integrated Matching for Picture LIbraries*. In IEEE Trans. on pattern Analysis and Machine Intelligence, volume 23, pages 947--963, 2001.
9. A. Vailaya, M. Figueiredo, A. Jain, and H. J. Zhang, "Content-Based Hierarchical Classification of Vacation Images," *Proceedings of IEEE International Conference on Multimedia Computing and Systems*, 1999.
10. C.P. Town and D. Sinclair. Content based image retrieval using semantic visual categories. Technical Report 2000.14, AT&T Laboratories Cambridge, 2000.
11. Ben Bradshaw. Semantic Based Image Retrieval: A Probabilistic Approach. *ACM Multimedia 2000*, Oct. 2000.
12. T. M. Mitchell. *Machine learning*. McGraw Hill, New York, US, 1996.
13. Stéphane Bissol, Philippe Mulhem, Yves Chiaramella, *Dynamic Learning of Indexing Concepts for Home Image Retrieval*, in Content-Based Multimedia Indexing (CBMI2003), Rennes (France), pp87-93, 22-24 septembre, 2003.